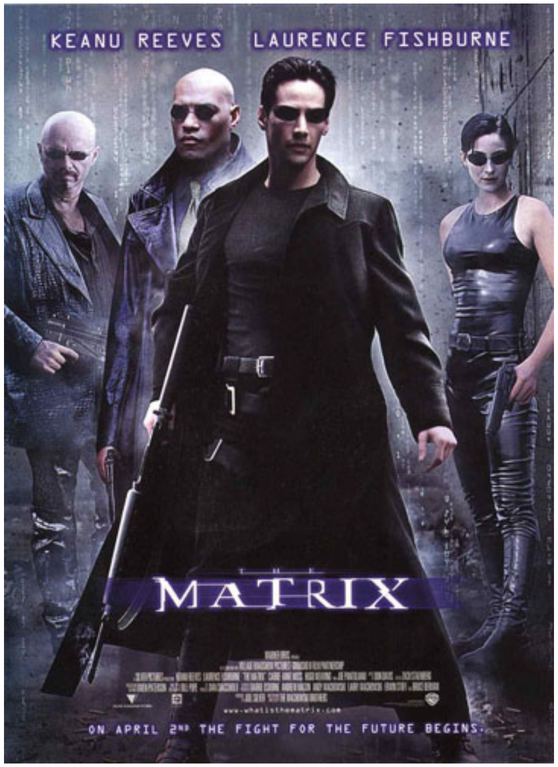
Virtual Machines in Condor

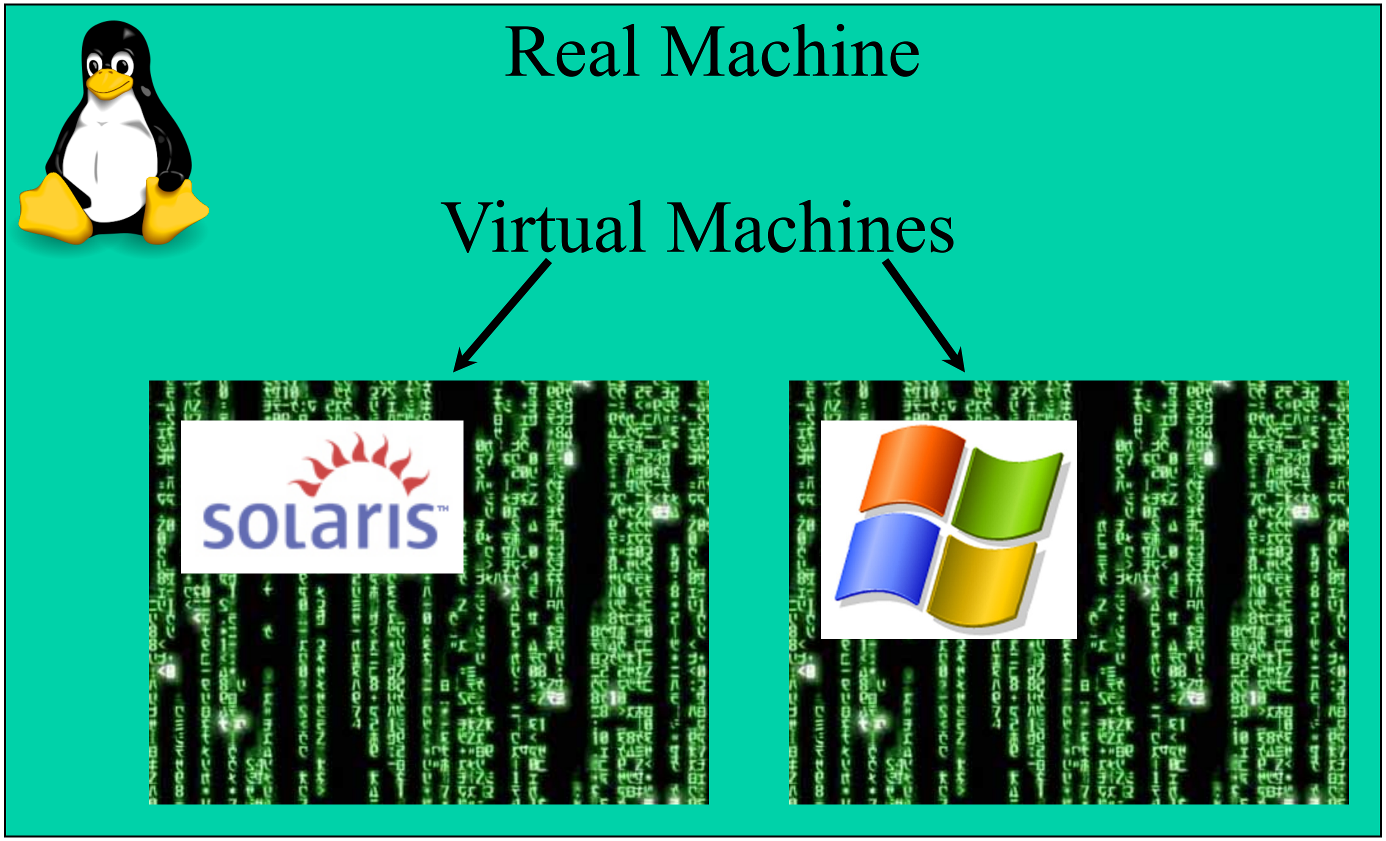
Condor Project
Computer Sciences Department
University of Wisconsin-Madison

Virtual Machines

- Simulated hardware
- Software in the VM thinks it's running on a normal machine



Virtual Machines



Real Machine

Virtual Machines

Benefits of Virtual Machines

- Job sandboxing
- Checkpoint and migration
- Jobs with elevated privileges
- Platform independence

Job Sandboxing

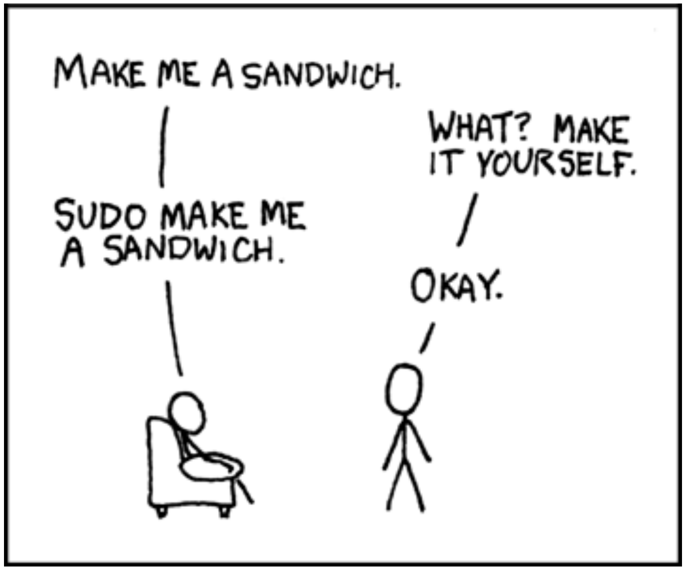
- Protect machines from jobs
  - Both accidental and malicious damage
- Machine owners more willing to run unfamiliar jobs

Checkpoint and Migration

- State of entire VM (OS and all) is recorded
- VM can be checkpointed for…
  - Failure recovery
  - Migration to other machines

Jobs with Elevated Privileges

- Run as root or administrator user
- Alter OS installation
- Useful for automated testing of software like Condor



Platform Independence

- Jobs can run on more machines
- Machines can run more jobs
- Linux jobs on Windows machines
  - And vice versa

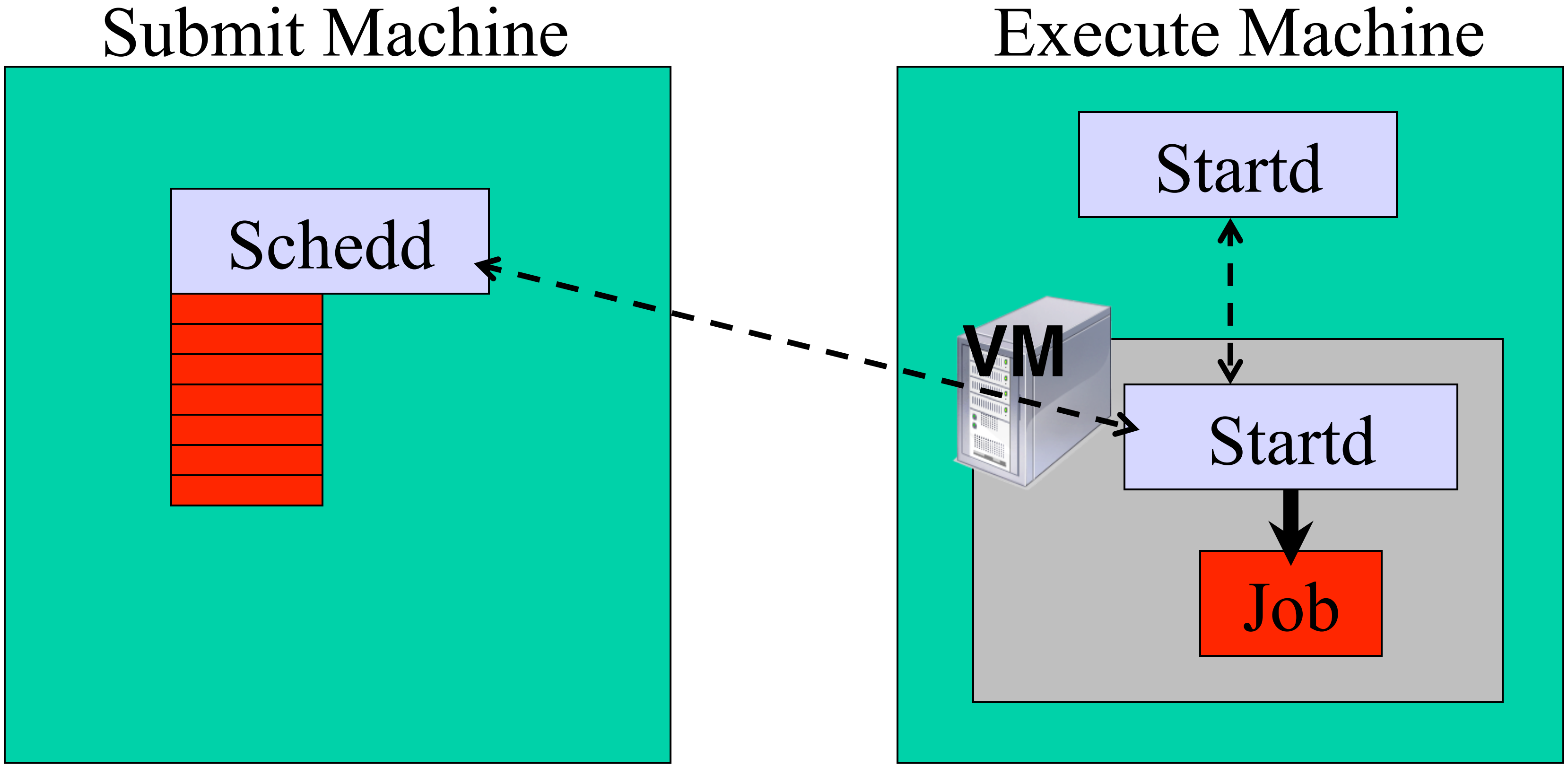
VM Image Provided By…

- Machine Owner
  - Condor runs inside a VM
  - VM becomes a node in your Condor pool
- Job Owner
  - VM universe
  - Condor runs a user-provided VM image

Condor in a VM

- Run Condor in a VM
- VM joins your pool
- VM acts like any other node
- Condor in VM can gather information from host machine
  - E.g. load average, keyboard idle time

Condor in a VM



Config Settings

Host config file

```
VMP_VM_LIST = vm1.bar.edu, vm2.bar.edu
```

```
HOSTALLOW_WRITE = $(HOSTALLOW_WRITE), \
    $(VMP_VM_LSIT)
```

VM config file

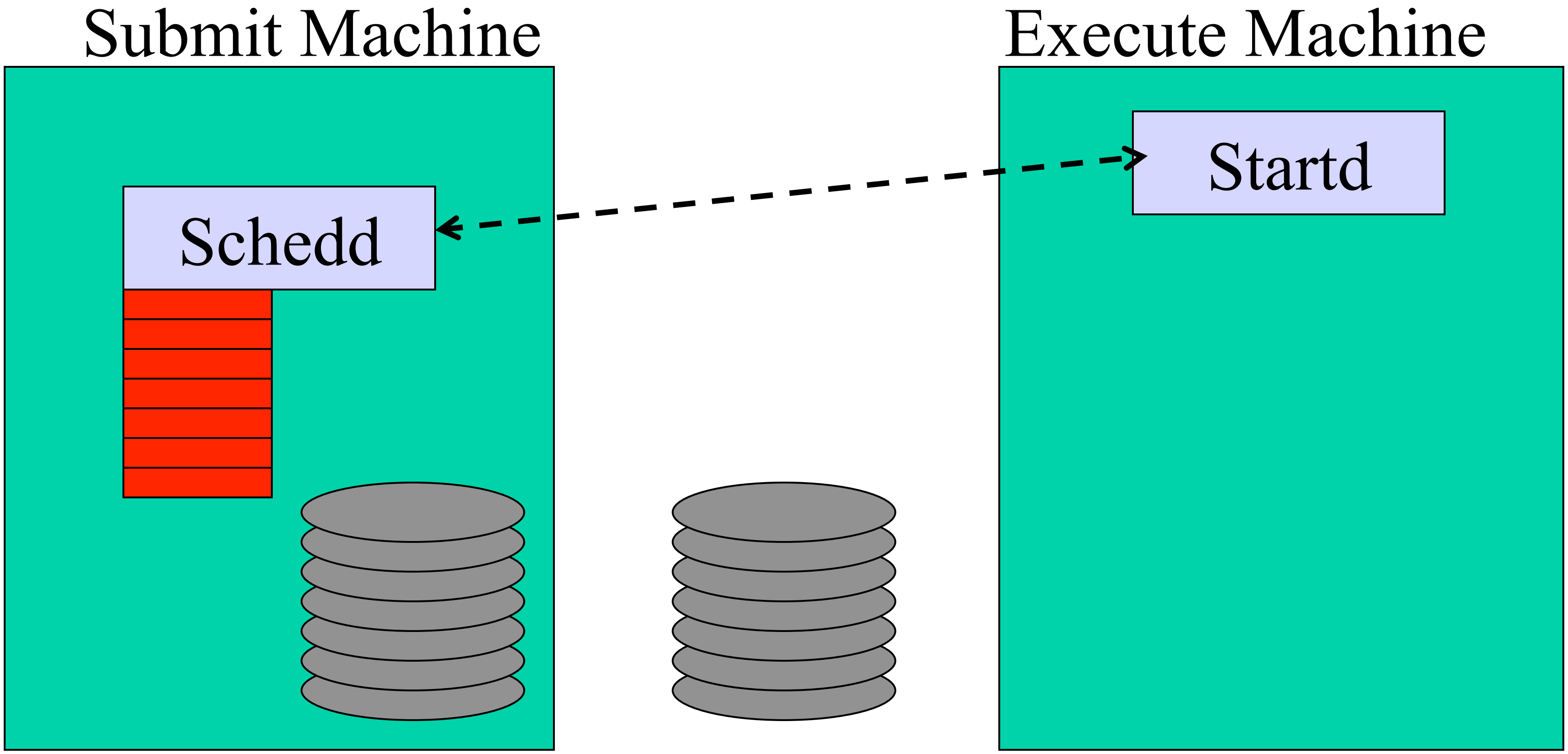
```
VMP_HOST_MACHINE = foo.bar.edu
```

```
START = (KeyboardIdle > 150) && \
    (HOST_KeyboardIdle > 150)
```

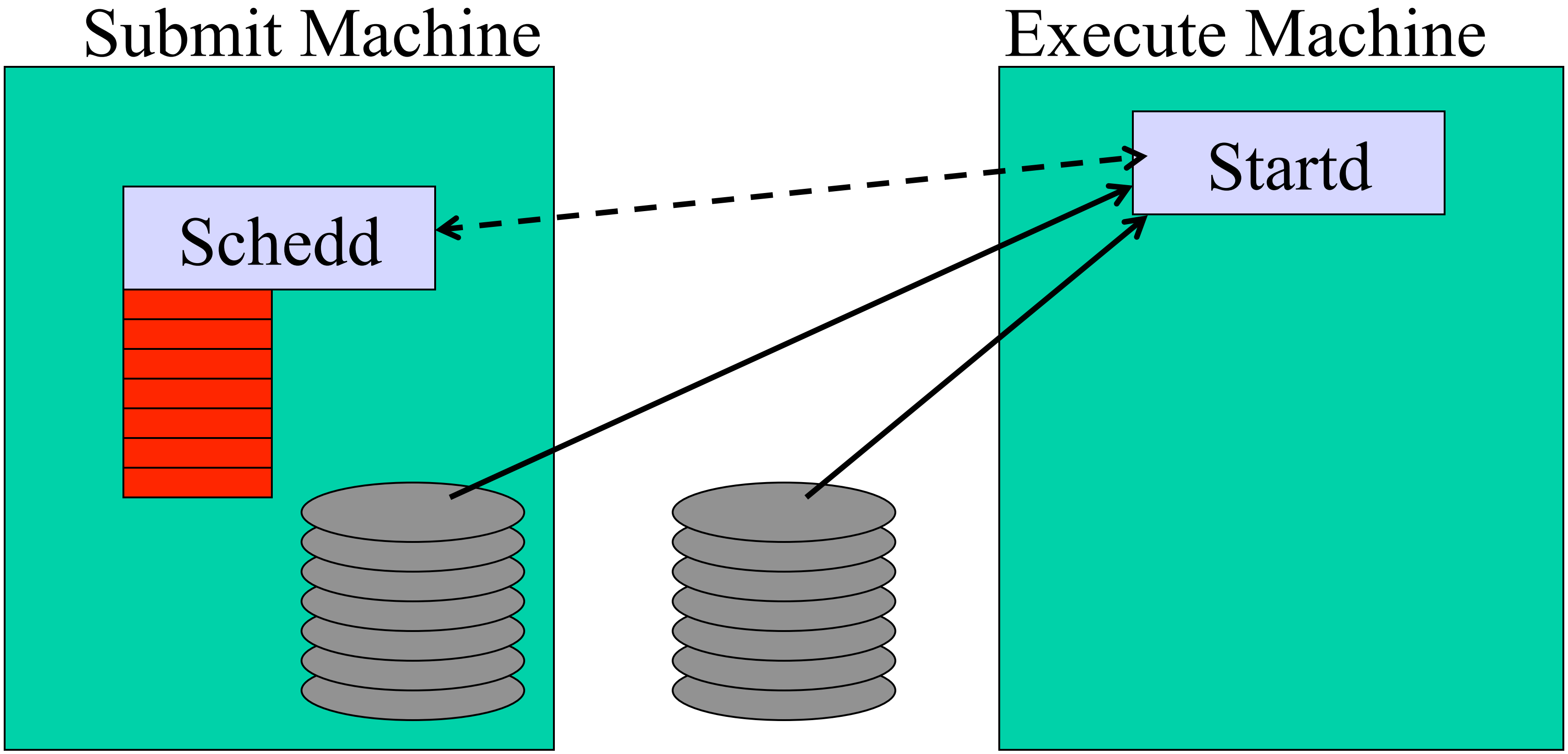
VM Universe

- The VM image is the job
- Job output is the modified VM image
- VMWare, KVM and Xen are supported
- VM GAHP
  - Daemon used to condor_starter to interact with VM software

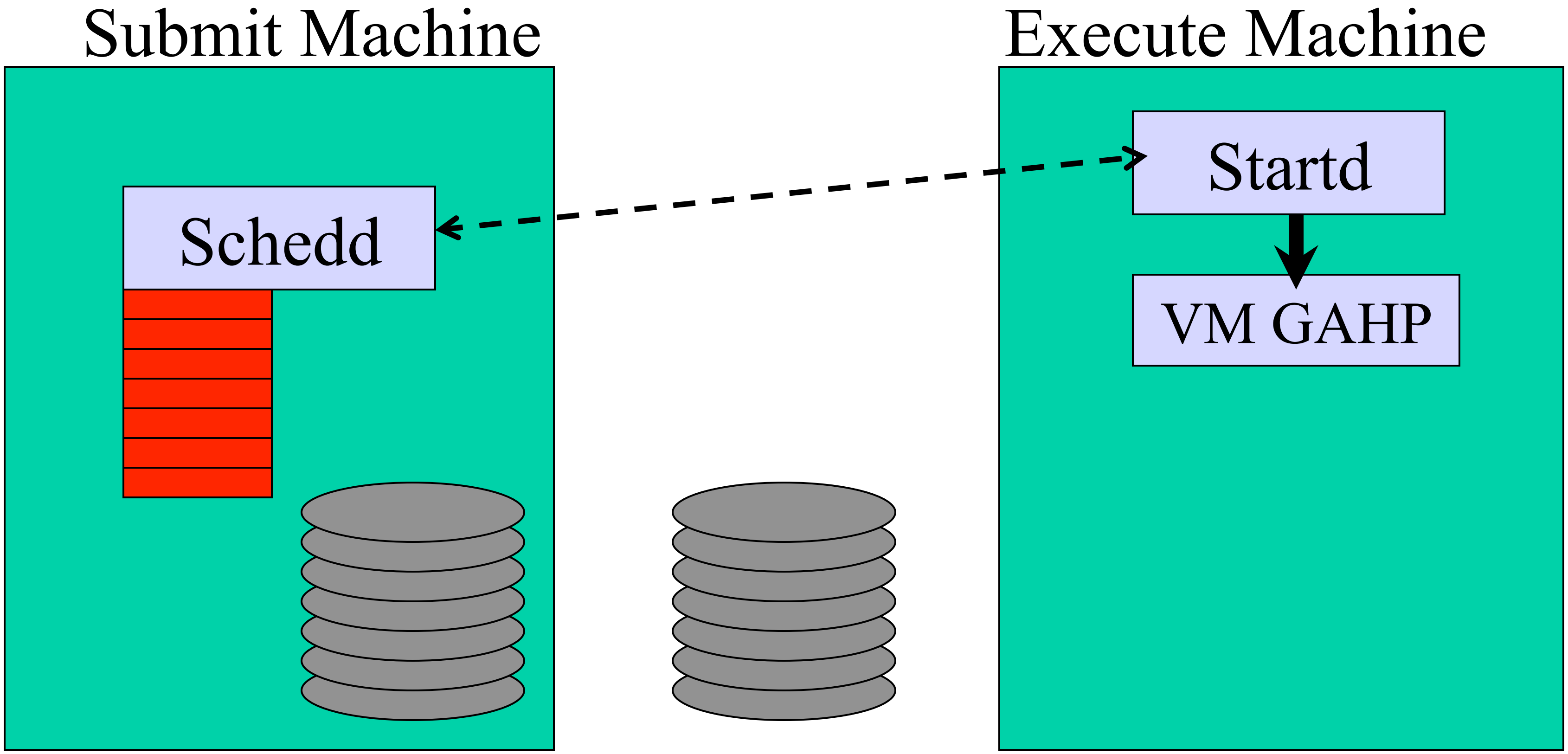
VM Universe Example



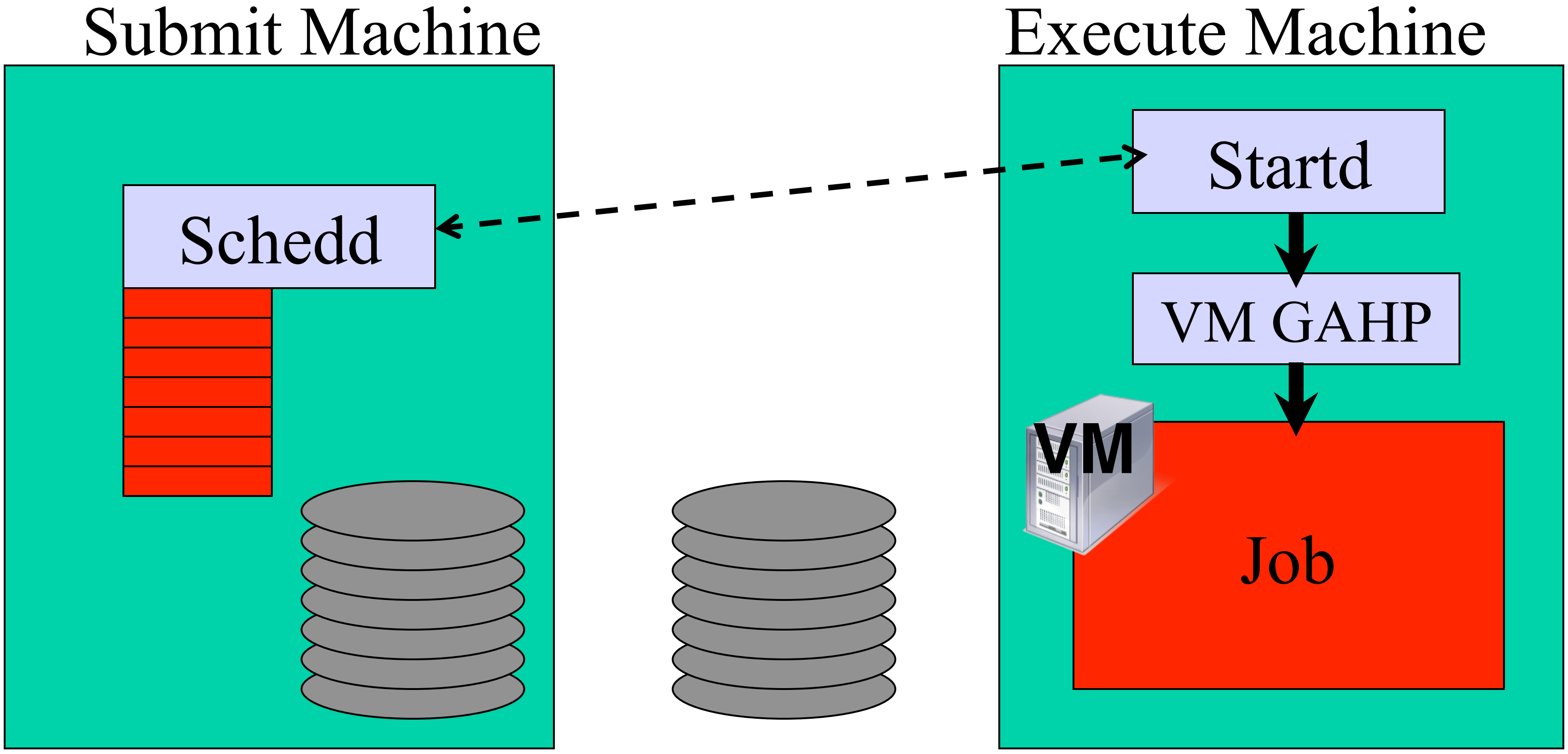
VM Universe Example



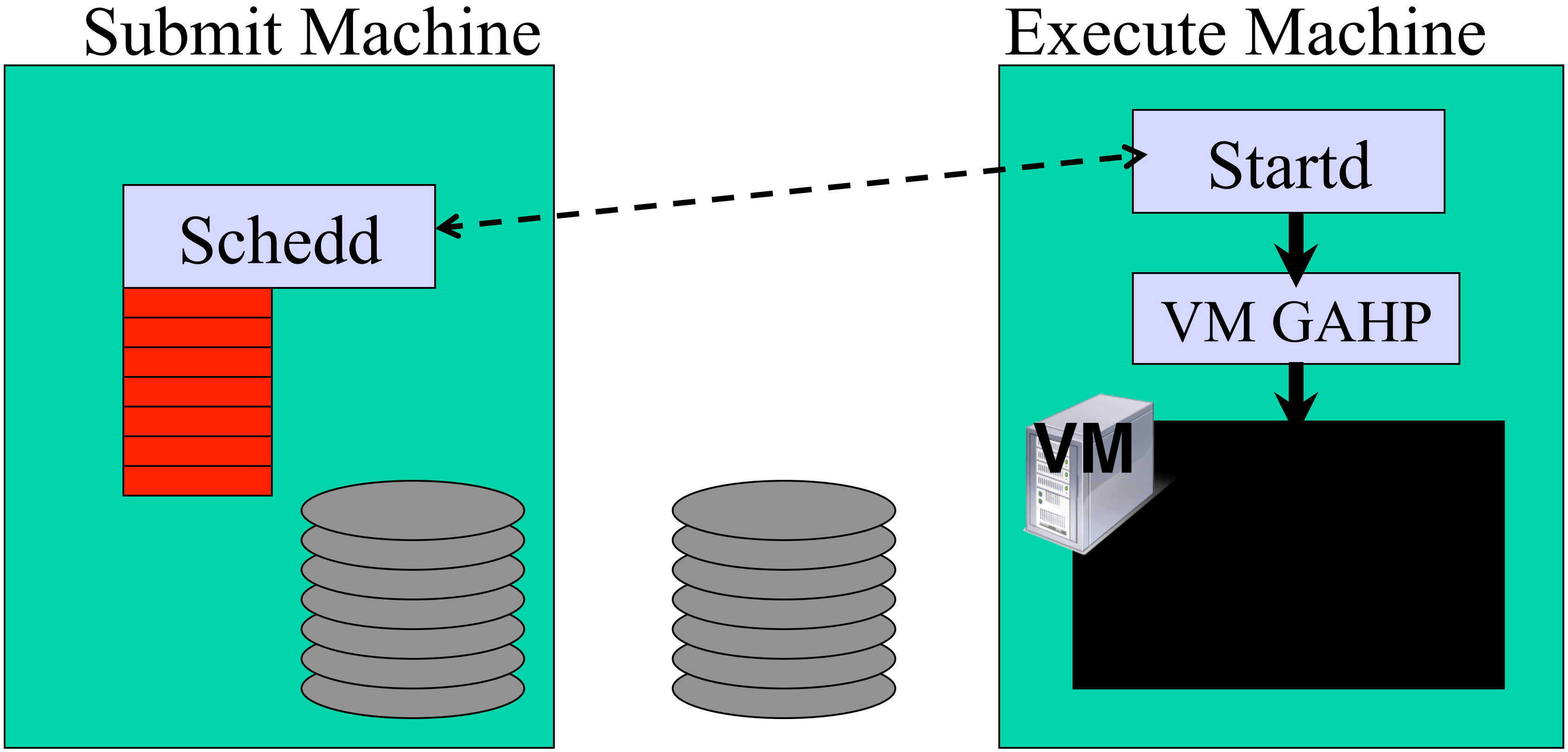
VM Universe Example



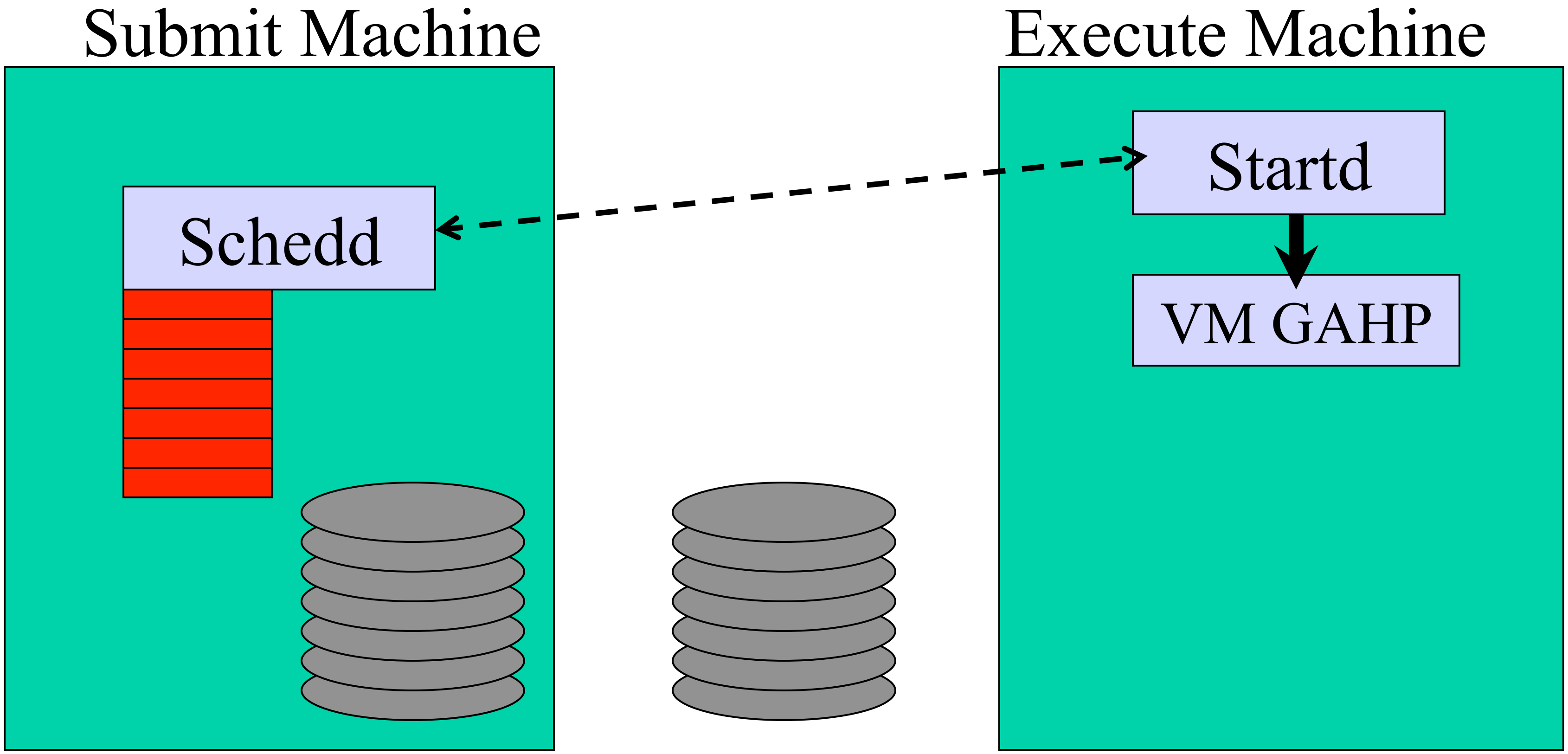
VM Universe Example



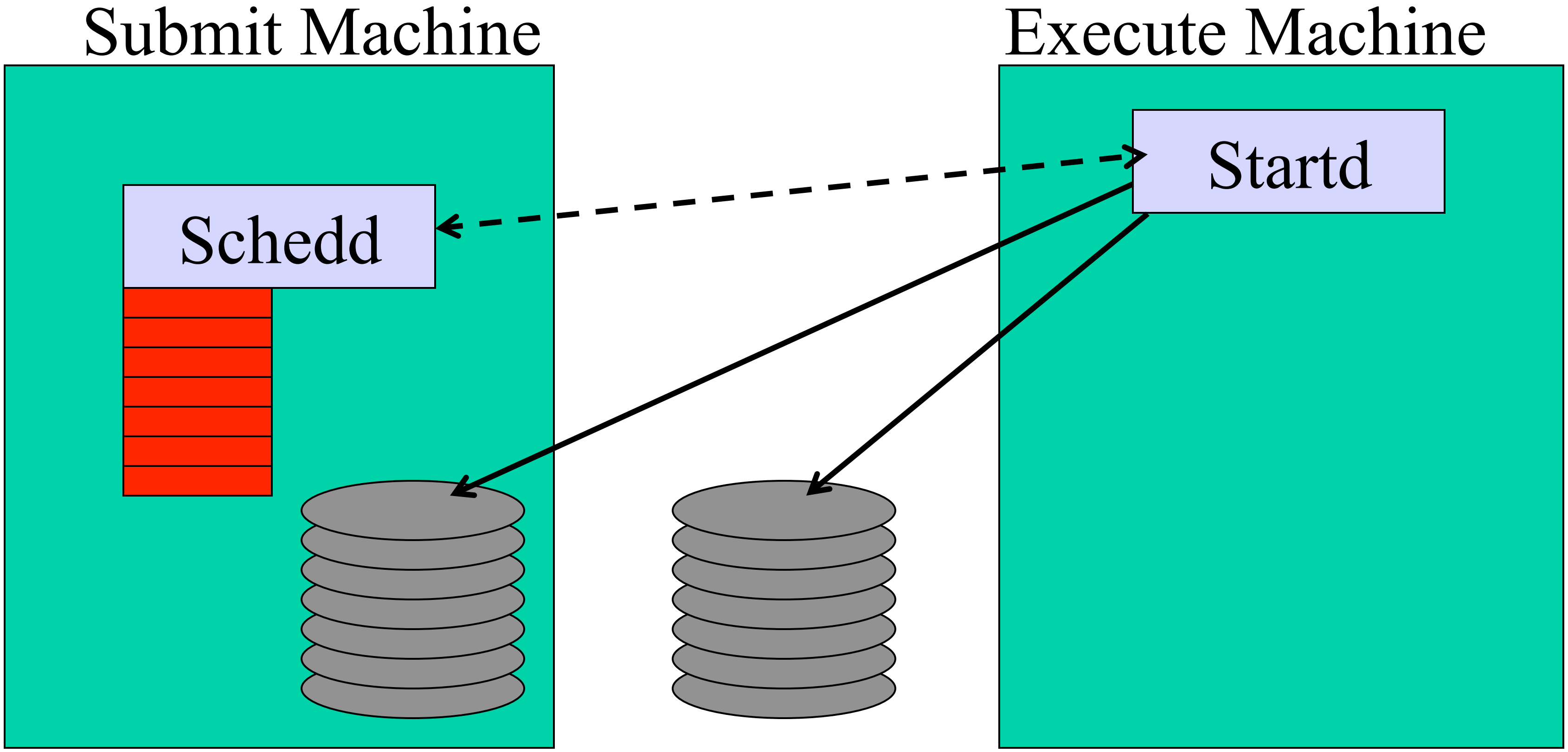
VM Universe Example



VM Universe Example



VM Universe Example



Condor Config File

- `VM_TYPE = <xen|kvm|vmware>`
  - Indicate what VM software you have
  - This enables VM capabilities
- `VM_MEMORY = 256`
  - Max memory all VMs can use
- `VM_MAX_NUMBER = 2`
  - Max simultaneous VMs

Condor Config File

- `VM_NETWORKING = TRUE`
  - Can the VM access the network?
- `VM_NETWORKING_TYPE = nat, bridge`
  - Ways the VM access the network
- `VM_NETWORKING_DEFAULT_TYPE = nat`
  - Default network access type
- `VM_SOFT_SUSPEND = True`
  - Suspend VM in memory or write to disk?

Config File for VMWare

- `VMWARE_NETWORKING_TYPE = \`
  `      <nat|bridged>`
  - Networking type to appear in .vmx file
- `VMWARE_LOCAL_SETTINGS_FILE = \`
  `      /path/to/file`
  - Extra attributes to insert in .vmx file

Config File for Xen/KVM

```
LIBVIRT_XML_SCRIPT = \
  $(LIBEXEC)/libvirt_simple_script.awk
```

- Optional callout to write libvirt XML description

```
VM_BRIDGE_SCRIPT = \
   vif-bridge bridge=xenbr0
```

- Script to set up networking

```
XEN_BOOTLOADER = /usr/bin/pygrub
```

- Xen only, when kernel included in disk image

Machine ClassAd

```
HasVM = True
VM_AvailNum = 2
VM_Memory = 256
VM_Networking = True
VM_Networking_Types = "nat,bridge"
VM_GAHP_VERSION = "$VMGahpVersion…"
VM_Type = "vmware"
```

Build a Submit File

- `universe = vm`
- `executable = MyJob1`
  - Executable only used for naming in condor_q display
- `vm_type = <vmware|kvm|xen>`

Build a Submit File

- `vm_memory = 256`
  - Units are megabytes

Build a Submit File

- `vm_networking = <True|False>`
  - Does VM require a network interface?
  - Some machines may not provide one
- `vm_networking_type = <nat|bridge>`
  - Does VM require a specific type of network interface?
  - Some machines may not provide both types

Build a Submit File

- `vm_no_output_vm = \`
  `<True|False>`
  - Should modified VM image be returned to user?
  - Some VM jobs may send results over the network

Build a Submit File

- `vm_cdrom_files = a.txt, b.txt`
  - Files are mounted in VM as a CD-ROM image
  - Allows you to use a VM image for many different jobs
  - You can replace the list of files with a single ISO image

Build a Submit File

- `vm_should_transfer_cdrom_files = \`
  `    <True|False>`
  - If True, files for CD-ROM image are transferred from submit machine to execute machine
  - If False, files are read from a shared filesystem on execute machine

Build a Submit File

- `vm_checkpoint = <True|False>`
  - If True, Condor will checkpoint VM periodically and on eviction from execute machine
  - Checkpoints stored on submit machine

VMWare Parameters

- `vmware_dir = <path>`
  - Directory containing the VMWare VM image to be run

VMWare Parameters

```
vmware_snapshot_disk = \
      <True|False>
```

- A snapshot disk records only the changes from the original VM image
- Saves network bandwidth and disk space on submit machine

VMWare Parameters

- `vmware_should_transfer_files = \`
  `<True|False>`
  - If True, files in `vmware_dir` are transferred from submit machine to execute machine
  - If False, files are read from a shared file system on execute machine

Xen/KVM Parameters

-
  ```
  xen_disk = file1:dev1:perm1,\
      file2:dev2:perm2
  ```

-
  ```
  kvm_disk = file1:dev1:perm1,\
      file2:dev2:perm2
  ```
  - The VM image is a list of disk image files, along with the devices they should be mapped to in the VM and the permissions they should have
  - The image files can be whole disks or disk partitions

Xen Parameters

- `xen_kernel = included`
  - The kernel is in the disk image file
- `xen_kernel =` *`/path/to/kernel`*
  - Use the indicated kernel

Xen Parameters

- `xen_kernel_params = <params>`
  - Append `<params>` to Xen kernel command line
- `xen_root = <device>`
  - Indicates root disk when kernel not included in disk image
- `xen_initrd = <path>`
  - Path to ramdisk image to be used

Xen/KVM Parameters

- `xen_cdrom_device = <device>`
- `kvm_cdrom_device = <device>`
  - When using `vm_cdrom_files`, you must specify what device the CD-ROM image will be mapped to

Xen/KVM Parameters

- `xen_transfer_files = file1, file2`
- `kvm_transfer_files = file1, file2`
  - Xen-related files to be transferred from the submit machine to the execute machine
  - Any VM image files not listed are assumed to accessible on the execute machine

Checkpointing and Networking

- VM's MAC and IP address are saved across checkpoint and restart
- Network connections may be lost
  - If NAT networking is used and job changes machines
  - If job is idle for too long before restart
- VMWare provides a tool to maintain DHCP leases across checkpoint and restart

VM Checkpointing vs. Standard Universe

- No relinking
- Works with more types of jobs
  - Multiple processes and threads
  - Networking (but migration problematic)
- No Remote IO
  - Must specify input files

Creating a VM Image

- Configure OS to…
  - Run your application on boot-up
  - Shut down when your application exits
- Input files can be read from CD-ROM image
  - Input files can include application binary

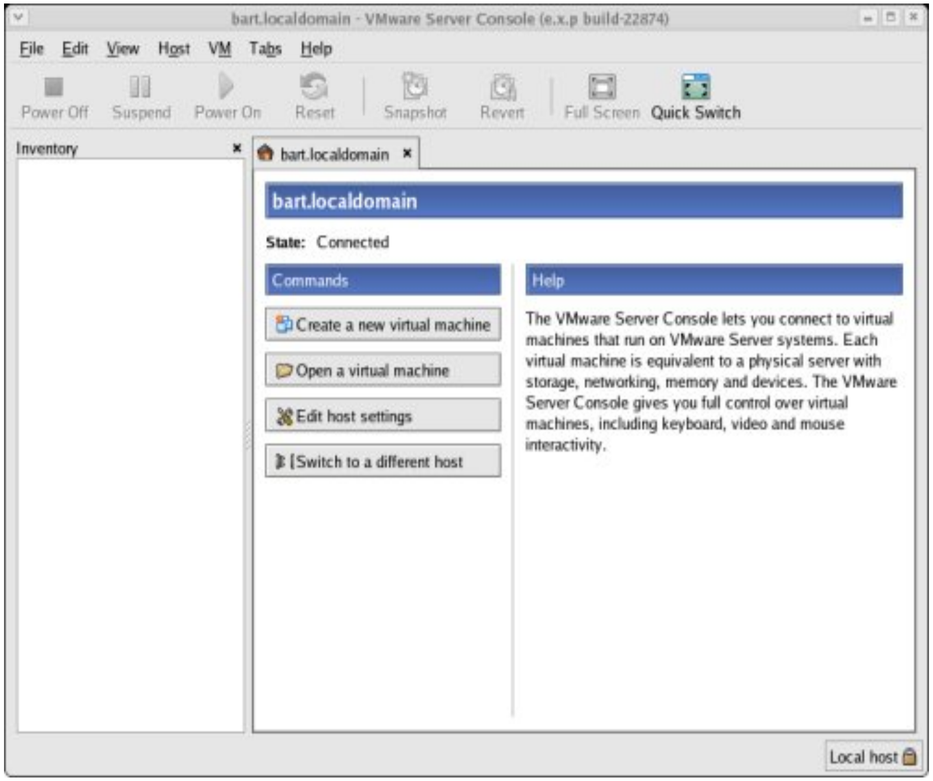
Running in the VM

- Sample boot script on linux
  - /etc/rc.d/rc3.d/S90myjob:

```
#!/bin/sh
su - joe ~/myjob 123 >~/output
shutdown -h now
```

How to Create VM images

- VMware Server
    - Using VMware Server Console



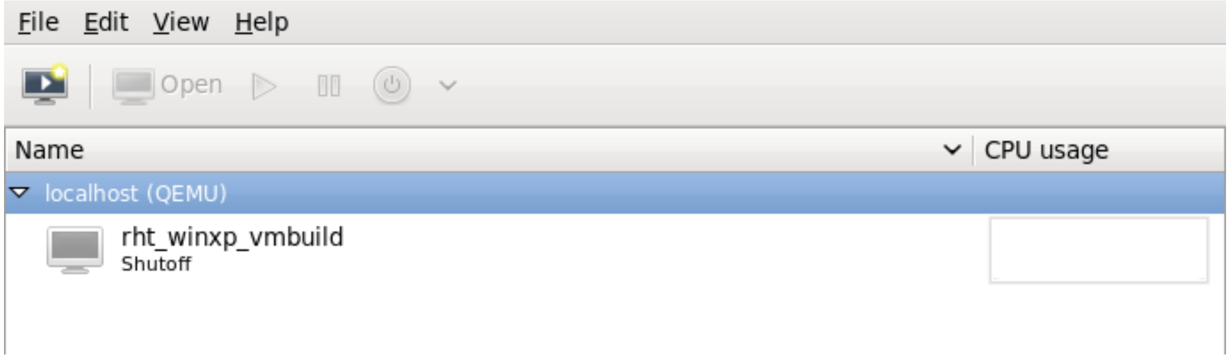
The VMware Server Console lets you connect to virtual machines that run on VMware Server systems. Each virtual machine is equivalent to a physical server with storage, networking, memory and devices. The VMware Server Console gives you full control over virtual machines, including keyboard, video and mouse interactivity.

How to Create VM images

- VMware Server
  - Can download pre-created VMs from http://www.vmware.com/appliances/
  - Many Linux distributions: Ubuntu, Fedora, Red Hat Enterprise, openSUSE, CentOS

How to Create VM images

- Xen and KVM
  - Several Linux distributions have GUI or command line tool to create a VM image
    - On Fedora Core, virt-install and virt-manager
    - On OpenSuse, through YaST
  - Can create a VM from scratch by using dd, mke2fs, and mount –o loop



Name | CPU usage

localhost (QEMU)

rht_winxp_vmbuild
Shutoff

Small VM Images

- Damn Small Linux
  - www.damnsmalllinux.org
  - As small as 6MB
- LitePC
  - www.litepc.com
  - Windows 2000 in 150MB
  - Windows 9x in 40MB

Thank You

- Any questions?
- Several VM-related talks on Wednesday
- Discussion: Virtual Machines and Condor
  - Friday, 11:30-12:15